UNIT -3

SOFTWARE TESTING

CHAPTERS

- SIX ESSENTIALS OF SOFTWARE TESTING
- STATE OF ART AND STATE OF PRACTICE
- CLEAN SHEET APPROACH
- ESTABLISHING A PRACTICAL PERSPECTIVES
- CRITICAL CHOICES
- CRITICAL DISIPLINES
- FRAMEWORKS FOR TESTING

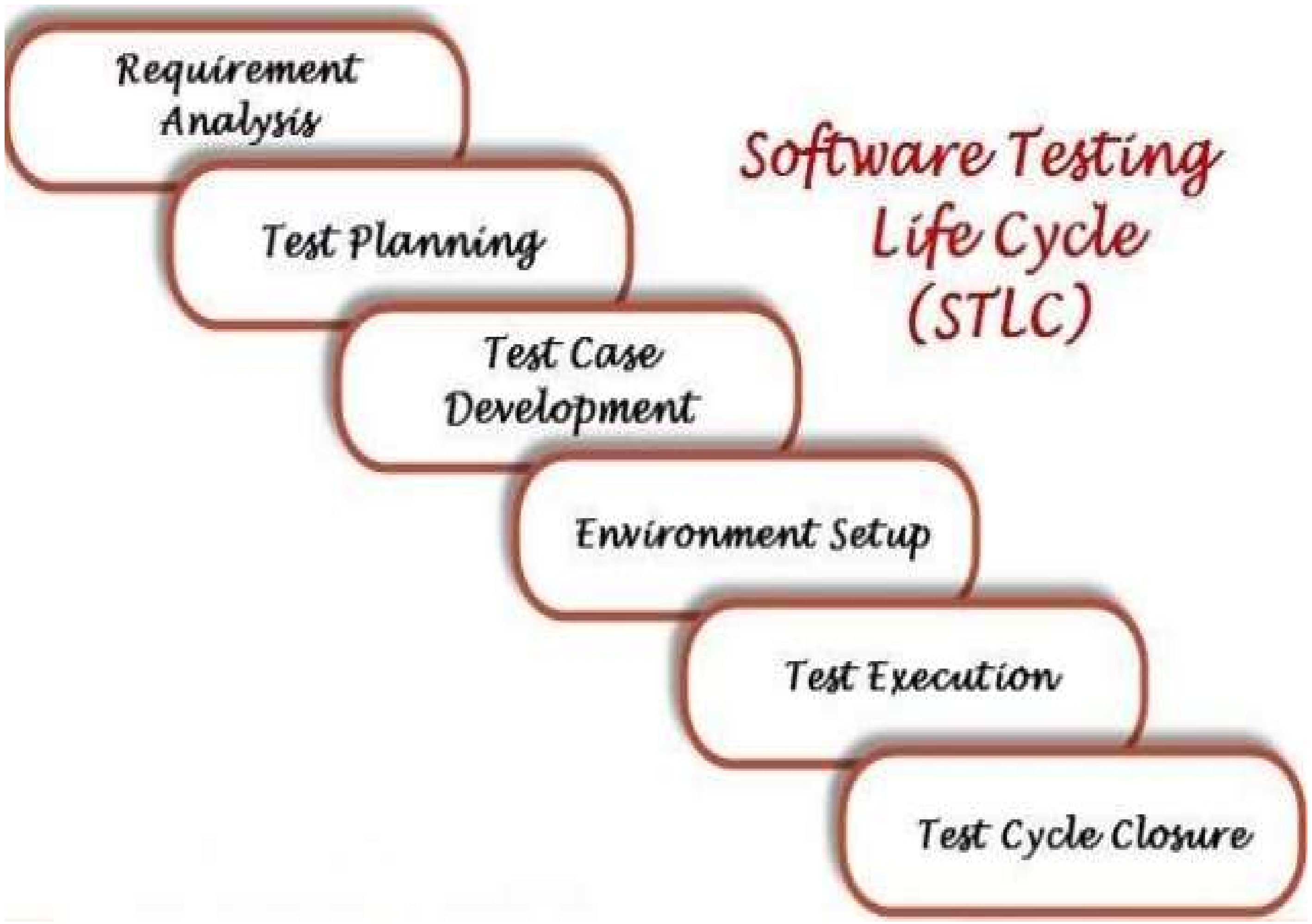
- SIX ESSENTIALS OF SOFTWARE TESTING

1. **THE QUALITY OF THE TEST PROCESS DETERMINES THE SUCCESS OF THE TEST EFFORT**
2. **PREVENT DEFECT MIGRATION BY USING EARLY LIFE-CYCLE TESTING TECHNIQUES**
3. **THE TIME FOR SOFTWARE TESTING TOOLS IS NOW**
4. **A REAL PERSON MUST TAKE RESPONSIBILITY FOR IMPROVING THE TESTING PROCESS**
5. **TESTING IS A PROFESSIONAL DISCIPLINE REQUIRING TRAINED, SKILLED PEOPLE**
6. **CULTIVATE A POSITIVE TEAM ATTITUDE OF CREATIVE DESTRUCTION**

ESSENTIAL 1 –

- **THE QUALITY OF THE TEST PROCESS DETERMINES THE SUCCESS OF THE TEST EFFORT**
- The quality of software system is primarily determined by the **quality of software process that produces it.**
- Testing has its **own cycle.**

  -Begin with product requirement phase and from there entire development process.
- Each phase is important testing activity.
- **Test groups that operate within the organizations** having an immature development process will feel more pain than those that don't.
- The test group should focus on improving it's own internal process.
- Immature test will result in unproductive ,chaotic, frustrating environment produces **low quality result and unsatisfactory result**.


Requirement Analysis
Test Planning
Test Case Development
Environment Setup
Test Execution
Test Cycle Closure
Software Testing Life Cycle (STLC)

ESSENTIAL 2

- **<u>PREVENT DEFECT MIGRATION BY USING EARLY LIFE-CYCLE TESTING TECHNIQUES</u>**
- The cost of the errors is minimized if they are detected in the same phase as they are introduced.
- An effective test program prevents the migration of errors from any development phase to any subsequent phases.

- **ESSENTIAL 3**
- **THE TIME FOR SOFTWARE TESTING TOOLS IS NOW**
- Wide variety of tools which may have mature and healthy product.
- Test-cycle **time reductions and automation providing for 24** hours a day of unattended test operations can be achieved with **capture and playback tools.**
- **Structural coverage tool** is used to determine if the software has been thoroughly tested, this tool tells us specifically **which parts of the product have in fact been** executed by our tests.
- It is no longer acceptable to expect customers to be the first to execute our code.

- **ESSENTIAL 4**
- **A REAL PERSON MUST TAKE RESPONSIBILITY FOR IMPROVING THE TESTING PROCESS**
- If the testing group is feeling pain, start campaigning for improvements to a few of the key issues, such as **better specifications, better reviews and inspections.**
- Management **should appoint architect** or core team to priotize the potential improvements
- It is not rocket science **but take take time and effort.**
- **Testing tools** helps but must be used with in overall testing process.
- Software testing is a process that requires people who take responsibility for its improvement.

- **ESSENTIAL 5**
- **<u>TESTING IS A PROFESSIONAL DISCIPLINE REQUIRING TRAINED, SKILLED PEOPLE</u>**
- The software testing, process has evolved considerably and has reached the point where it is a discipline **requiring trained professionals.**
- It has become career choice.
- It is not entry level job.
- When testing is done properly, it surpasses the challenge of product development.

- **ESSENTIAL 6**
- **CULTIVATE A POSITIVE TEAM ATTITUDE OF CREATIVE DESTRUCTION**
- Testing require disciplined creativity.
- Good testing is that which discover **defect in product** require real ingenuity, may be view as destructive.
- Establishing the proper **"test to break"** mental attitude has a profound effect on testing success.
- Testers make **vital positive contributions** throughout the development process to ensure the quality of the product

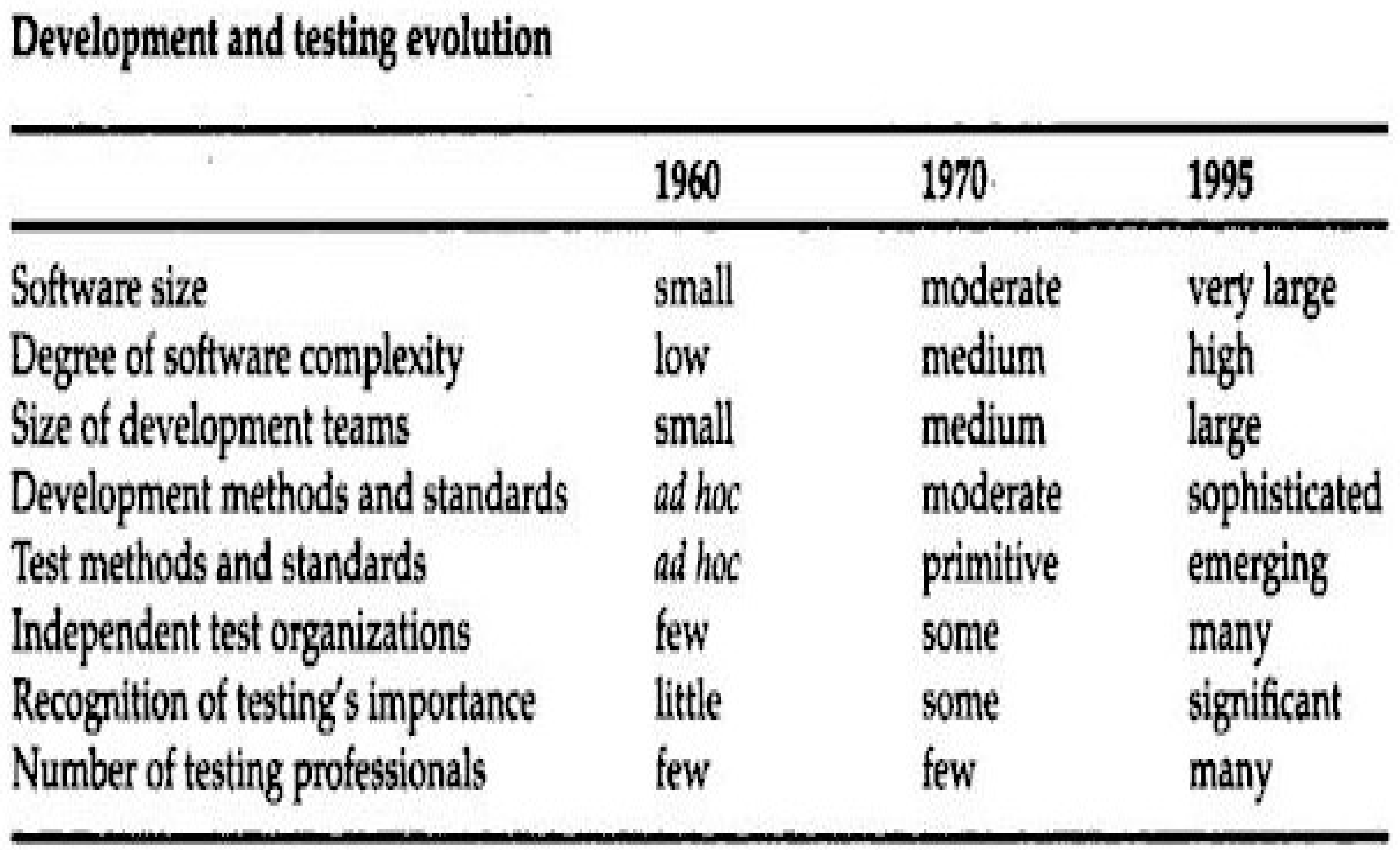
The state of the art and state of the practice

- Software testing is yet to become fundamental component of university software engineering curicula
- Many well known **methods are not used in industry today** and development of software system remain inordinately expensive, fraught with costly error
- **50% of development effort is spent on testing**
- Many of the people responsible for testing are **struggling for viable working platform , available tools in the software development environment.**

history

- Until **1956** it was the debugging oriented period, where testing was often **associated to debugging: there was no clear difference between testing and debugging.**
- From **1957-1978** there was the demonstration oriented period where **debugging and testing was distinguished** now - in this period it was shown, that software satisfies the requirements.
- The time between **1979-1982** is announced as the **destruction oriented period**, where the goal was to find errors.

- **1983-1987** is classified as the **evaluation oriented period**: intention here is that during the **software lifecycle -> product evaluation is provided to measure quality.**
- From **1988** on it was seen as **prevention oriented** period where tests were to demonstrate that software satisfies its **specification, to detect faults and to prevent faults.**

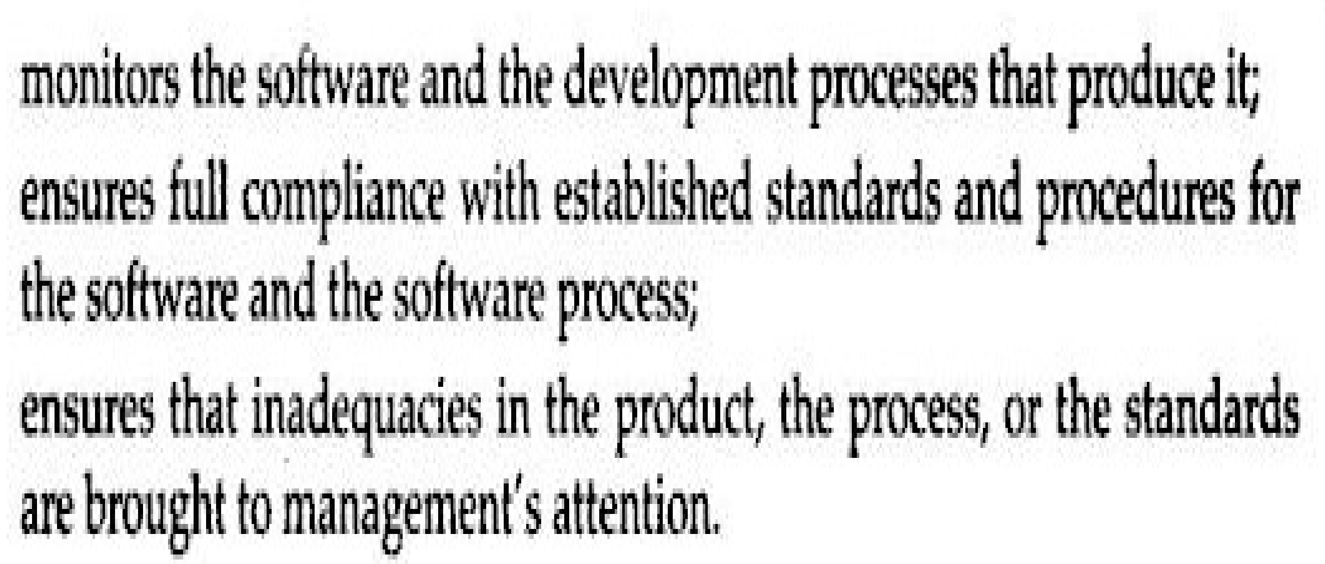


Development and testing evolution

	1960	1970	1995
Software size	small	moderate	very large
Degree of software complexity	low	medium	high
Size of development teams	small	medium	large
Development methods and standards	*ad hoc*	moderate	sophisticated
Test methods and standards	*ad hoc*	primitive	emerging
Independent test organizations	few	some	many
Recognition of testing's importance	little	some	significant
Number of testing professionals	few	few	many

SOFTWARE QUALITY ASSURANCE (SQA) is a set of activities for ensuring quality of software products

- Quality assurance defined in the literature as function with



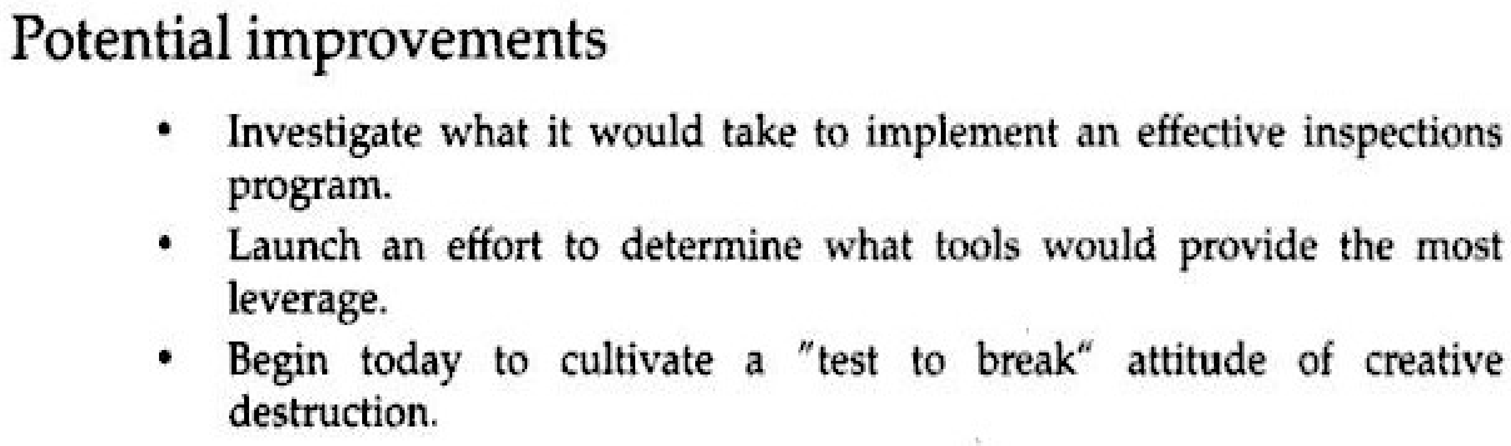
- monitors the software and the development processes that produce it;
- ensures full compliance with established standards and procedures for the software and the software process;
- ensures that inadequacies in the product, the process, or the standards are brought to management's attention.

The clean sheet approach to getting started

- Start with clean sheet of paper and **keep running list of potential improvement are key to renovating the software testing process.**

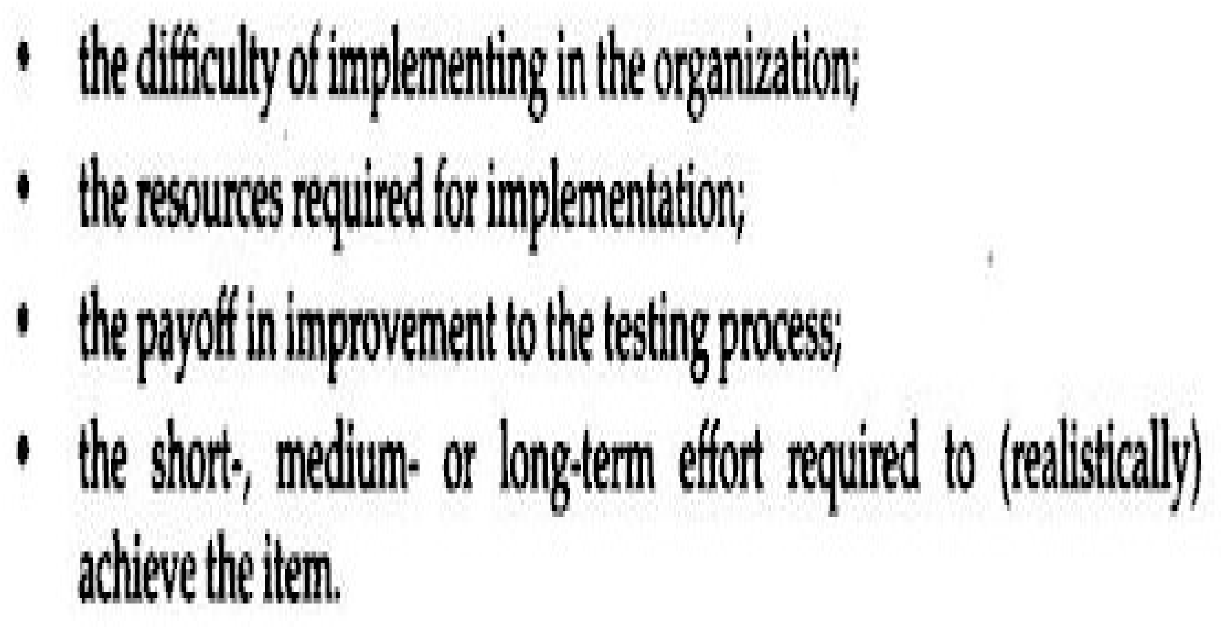
The idea take many forms like

- **New tools or tool you want to investigate**
- **Fundamental change in software test process**


Potential improvements

- Investigate what it would take to implement an effective inspections program.
- Launch an effort to determine what tools would provide the most leverage.
- Begin today to cultivate a "test to break" attitude of creative destruction.

- Prioritize improvements the software testing function group.
- Identify idea on your sheet as being



- the difficulty of implementing in the organization;
- the resources required for implementation;
- the payoff in improvement to the testing process;
- the short-, medium- or long-term effort required to (realistically) achieve the item.

Establishing a practical perspective

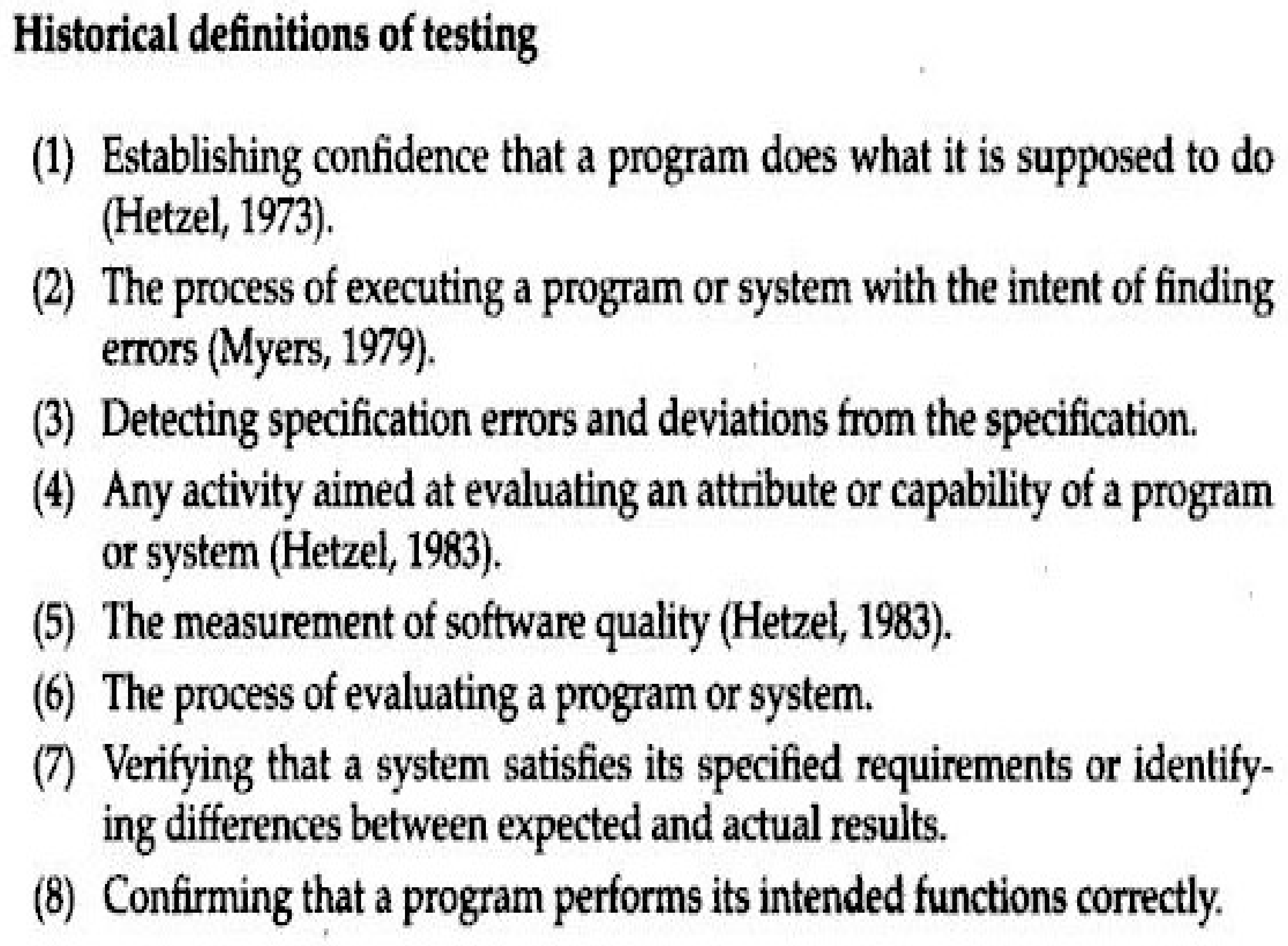
- Software is written by people and people make mistake.
- In standard commercial software errors are present.
- These errors are expensive
- **Work to locate them early, especially more critical one.**
- Implement appropriate testing technique
- Testers are not in charge of whole quality program .But quality decision should ultimately based on customer satisfaction.

- *Mistake*: A human action that produces an incorrect result.
- *Fault*: An incorrect step, process, or data definition in a computer program. The outgrowth of the mistake. (Potentially leads to a failure.)
- *Failure*: An incorrect result. The result (manifestation) of the fault (e.g., a crash).
- *Error*: The amount by which the result is incorrect.

Errors

- All error are costly: Undetected as well as **error detected late in the software development process are most expensive**
- Undetected error migrate downstream within system to cause failure.
- If it is detected only after later stages of the development they became costly to rework.
- If not detected cause failure.

What is testing really? Some definition



Historical definitions of testing

(1) Establishing confidence that a program does what it is supposed to do (Hetzel, 1973).
(2) The process of executing a program or system with the intent of finding errors (Myers, 1979).
(3) Detecting specification errors and deviations from the specification.
(4) Any activity aimed at evaluating an attribute or capability of a program or system (Hetzel, 1983).
(5) The measurement of software quality (Hetzel, 1983).
(6) The process of evaluating a program or system.
(7) Verifying that a system satisfies its specified requirements or identifying differences between expected and actual results.
(8) Confirming that a program performs its intended functions correctly.

The IEEE/ANSI definitions of testing

(1) The process of operating a system or component under specified conditions, observing or recording the results, and making an evaluation of some aspect of the system or component (IEEE/ANSI, 1990 [Std 610.12-1990]).

(2) The process of analyzing a software item to detect the difference between existing and required conditions (that is, bugs) and to evaluate the features of the software items (IEEE/ANSI, 1983 [Std 829-1983]).

The best definition for the tester

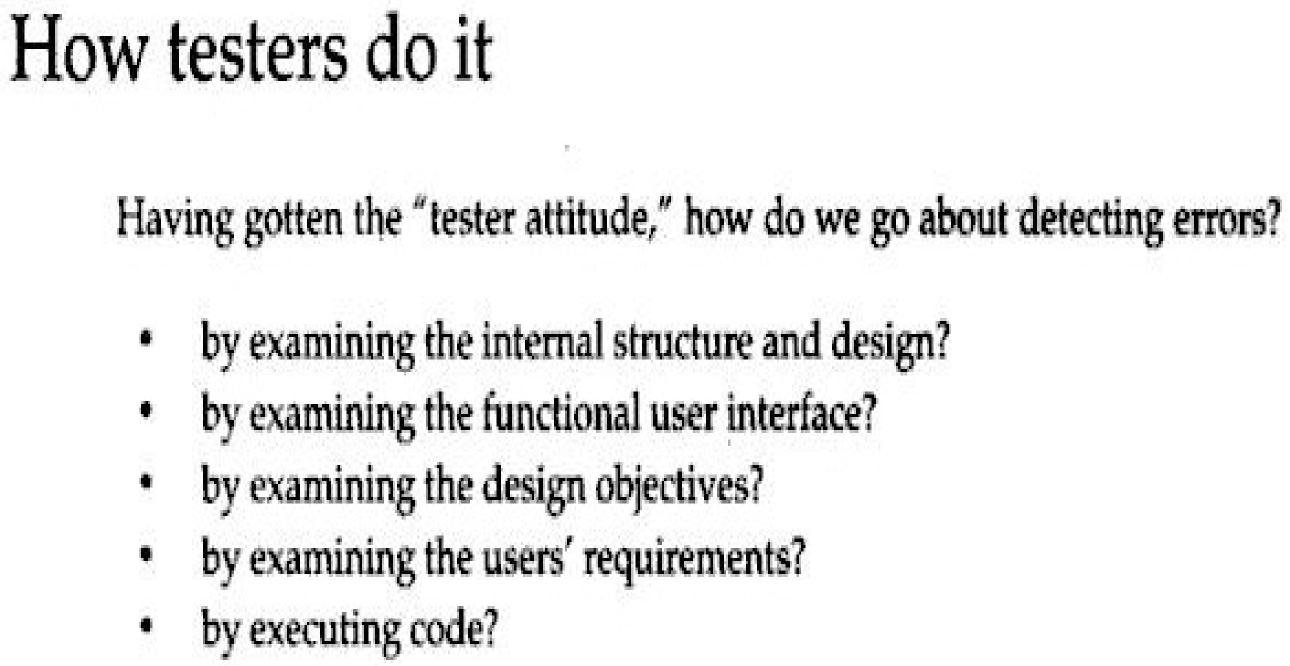
The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. (Myers, 1979)

Good testers have a testing attitude

Testers hunt errors

- If the tester job is to find every error or fault or weakness in the work product, then **good test is one that has good probability of detecting undiscovered error** and successful test is one that detect undiscovered error.
- The focus on showing the presence of error is basic attitude of good tester.
- **Detected error are celebrated for the good of the product**
- It gives personal satisfaction , we feel good if we find defects.

- **Testers are destructive- but creatively so**
- Testing is positive and creative effort of destruction .
- It take **imagination persistence and strong sense** of mission to systematically locate the weakness in complex structure and to demonstrate its failure.
- **Testers pursue error not people**
- Error are in the work product , not in the person who made mistake.
- With the test to destroy attitude we are not attacking individual in the organization or team but looking for error in the developed work product.
- The key to manage the developer and tester as team.
- **Testers add value**
- Testers add value to product by discovering error and getting them on table as early as possible

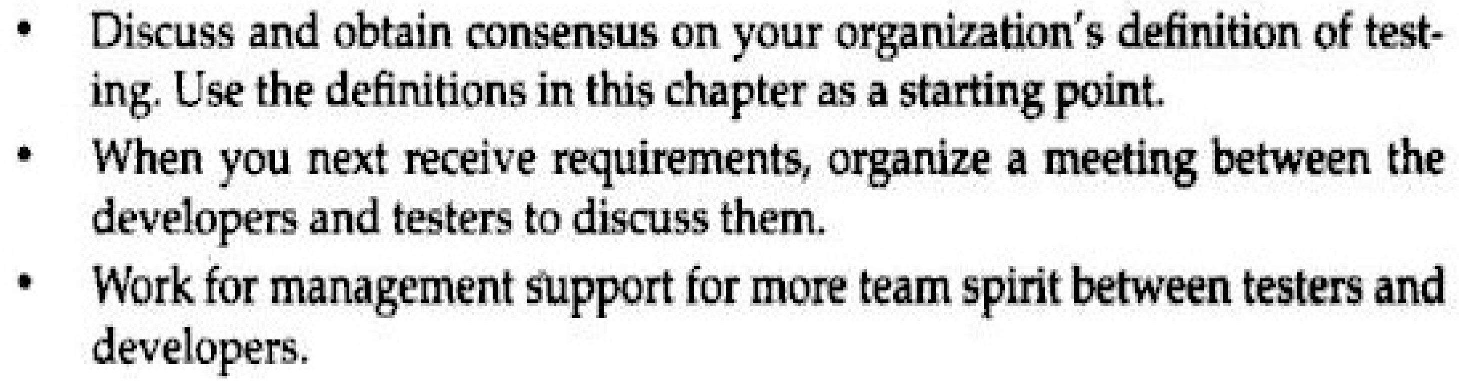


How testers do it

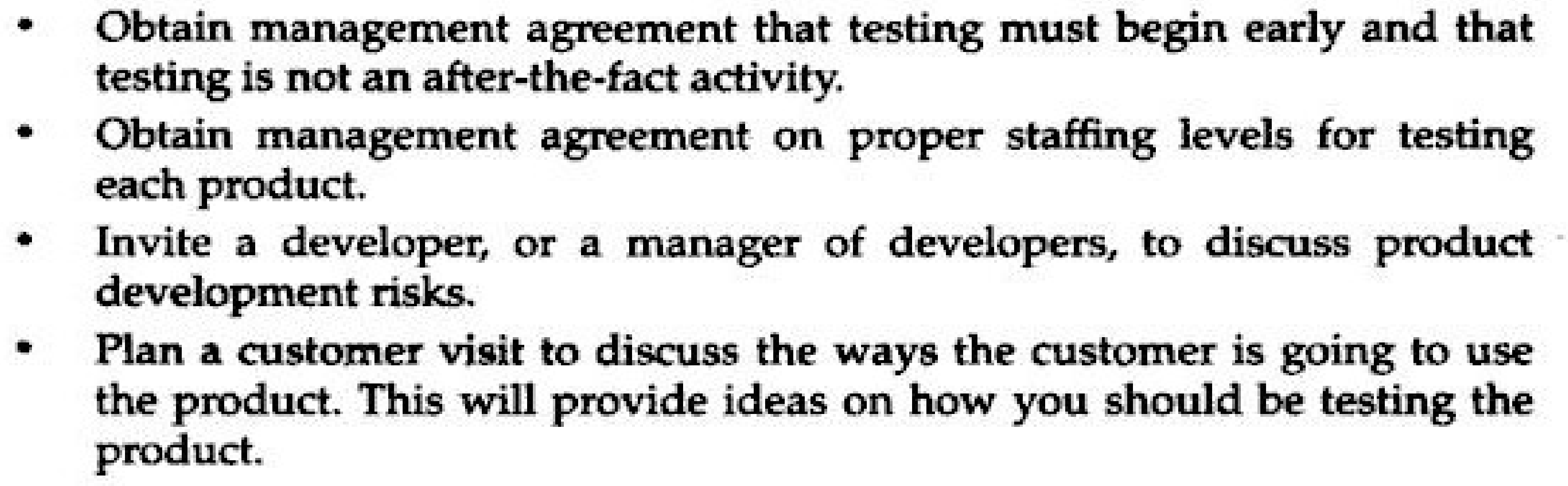
Having gotten the "tester attitude," how do we go about detecting errors?

- by examining the internal structure and design?
- by examining the functional user interface?
- by examining the design objectives?
- by examining the users' requirements?
- by executing code?

- **Suggestion for clean sheet approach:**



- Discuss and obtain consensus on your organization's definition of testing. Use the definitions in this chapter as a starting point.
- When you next receive requirements, organize a meeting between the developers and testers to discuss them.
- Work for management support for more team spirit between testers and developers.



- Obtain management agreement that testing must begin early and that testing is not an after-the-fact activity.
- Obtain management agreement on proper staffing levels for testing each product.
- Invite a developer, or a manager of developers, to discuss product development risks.
- Plan a customer visit to discuss the ways the customer is going to use the product. This will provide ideas on how you should be testing the product.